



# Detection of Rare Coins

Software Engineering Department

Braude College

Capstone Project Phase A – 61998

By Hassan Said Ahmad

Lev Leyfer

Supervisor: Zeev Frenkel

# Introduction

- Brief overview of rare coin detection.
- Importance of AI in solving this problem.
- Research objectives.



# Problem Statement

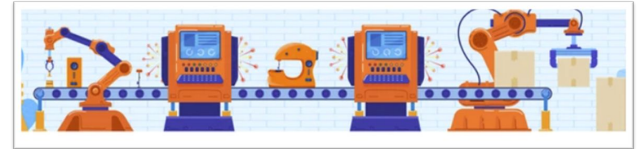
Challenges in rare coin identification.



Limitations of manual methods.



Need for an automated system.



# Proposed Solution

AI-based system overview

Model architecture

Key features



# Methods Used in Our Application

## Data Collection and Annotation:

- Collected 100 images of agorot coins.
- Annotated images to mark year stamps for training.



## Preprocessing:

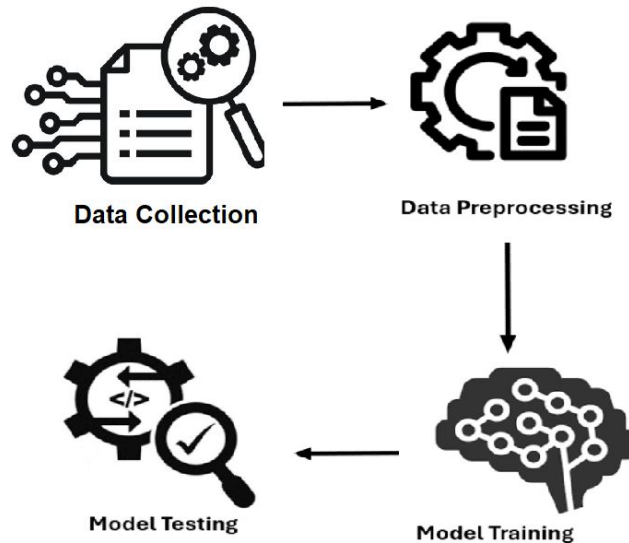
- Applied data augmentation (rotation, scaling, contrast adjustment).

## Model Training:

- Used the YOLO object detection framework.
- Fine-tuned pre-trained YOLO weights with our dataset.

## Evaluation and Testing:

- Measured model performance with precision, recall, and F1 score.
- Tested on unseen data for generalization.
- Conducted real-time testing via video feed.



# Converting Hebrew Year to Gregorian Year

## Hebrew Letters and Their Numerical Values:

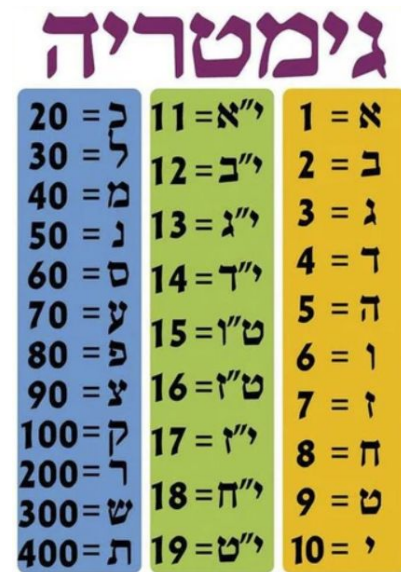
- The first nine letters represent the numbers 1 through 9:
  - $9 = \text{ט}$ ,  $8 = \text{ח}$ ,  $7 = \text{ז}$ ,  $6 = \text{ו}$ ,  $5 = \text{ה}$ ,  $4 = \text{ד}$ ,  $3 = \text{ג}$ ,  $2 = \text{ב}$ ,  $1 = \text{א}$
- The next nine letters represent the tens:
  - $90 = \text{צ}$ ,  $80 = \text{פ}$ ,  $70 = \text{ע}$ ,  $60 = \text{ס}$ ,  $50 = \text{נ}$ ,  $40 = \text{מ}$ ,  $30 = \text{ל}$ ,  $20 = \text{כ}$ ,  $10 = \text{י}$
- The remaining letters represent the hundreds:
  - $400 = \text{ת}$ ,  $300 = \text{ש}$ ,  $200 = \text{ר}$ ,  $100 = \text{ק}$

Example: תשפ"ד

- $400 = \text{ת}$
- $300 = \text{ש}$
- $80 = \text{פ}$
- $4 = \text{ד}$

Total:  $400+300+80+4=784$ . The thousands (5) are implied, so the year is 5784.

And now to make it into a civil year, we omit the first digit, and then we add 1240, in our example is  $784 + 1240 = 2024$



20 = כ	11 = י"א	1 = א
30 = ל	12 = י"ב	2 = ב
40 = מ	13 = י"ג	3 = ג
50 = נ	14 = י"ד	4 = ד
60 = ס	15 = י"ה	5 = ה
70 = ע	16 = י"ו	6 = ו
80 = פ	17 = י"ז	7 = ז
90 = צ	18 = י"ח	8 = ח
100 = ק	19 = י"ט	9 = ט
200 = ר		10 = י
300 = ש		
400 = ת		

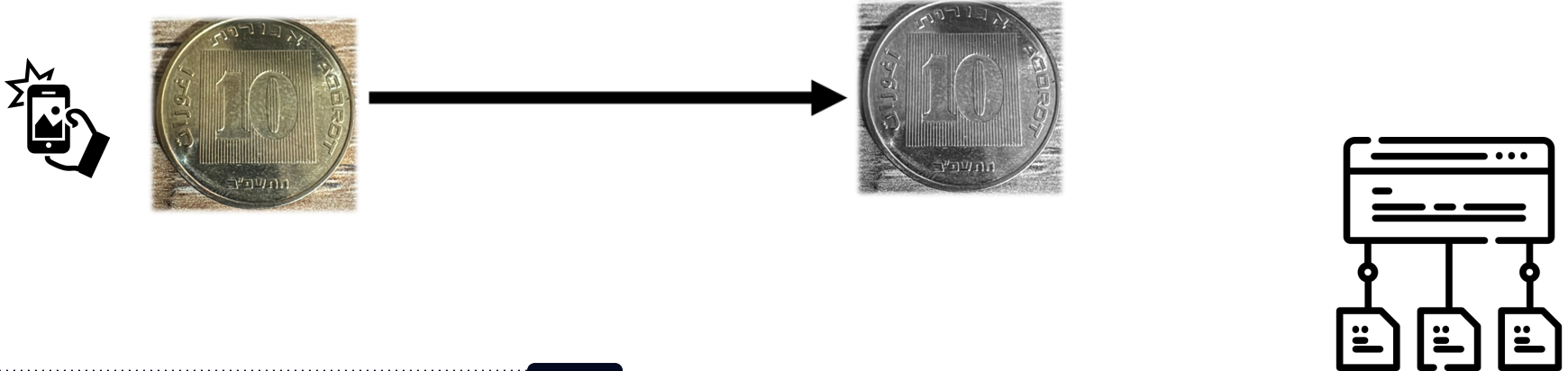
# Data Flow in the Coin Detection System

## Input:

- User captures an image or starts a video stream via the GUI.

## Pre-processing:

- Image or video is pre-processed for optimal detection.



## Coin Detection:

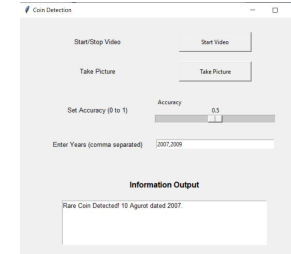
- Detection model identifies coins and extracts Hebrew letters stamped on them.

## Classification & Extraction:

- Coins are classified by rarity, mint year, and other metadata.
- Hebrew year is converted to civil year using the previously explained algorithm.

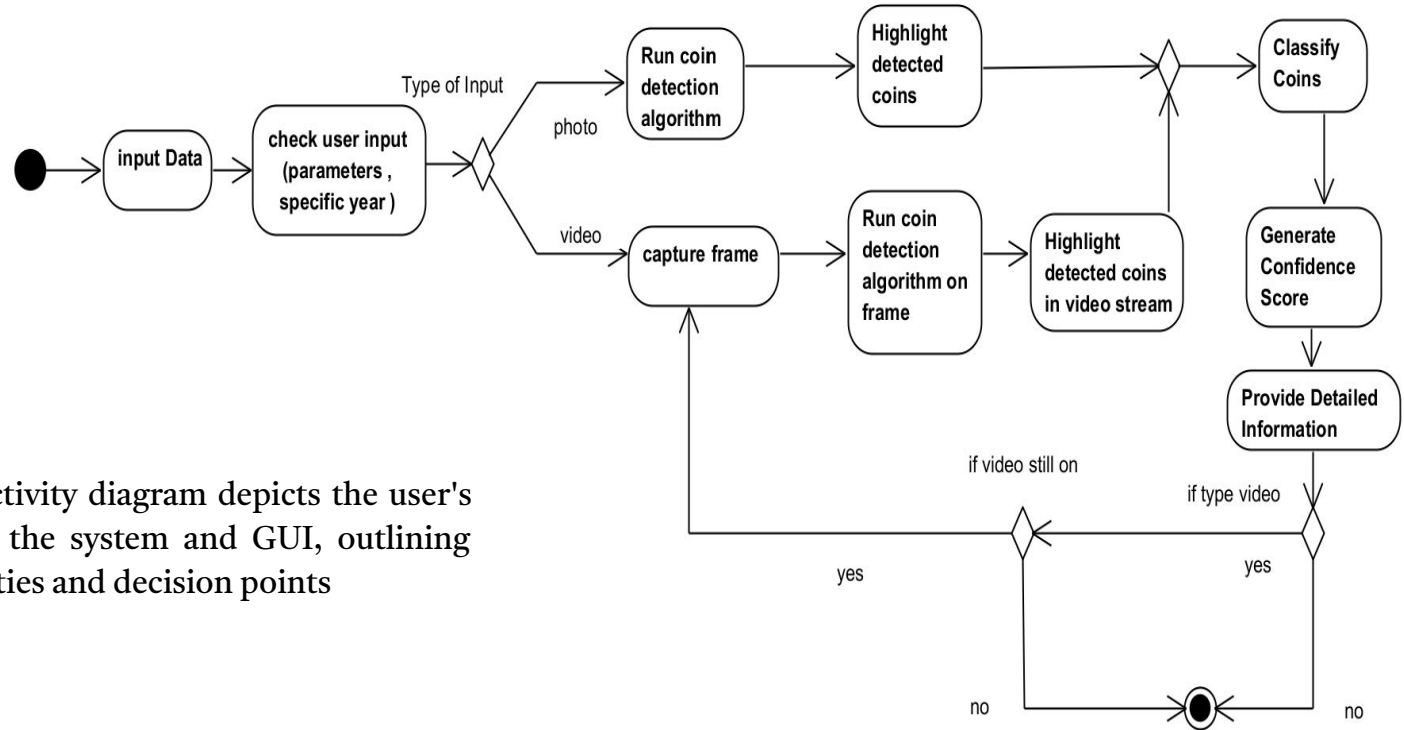
## Output:

- Results are displayed in the console or GUI.





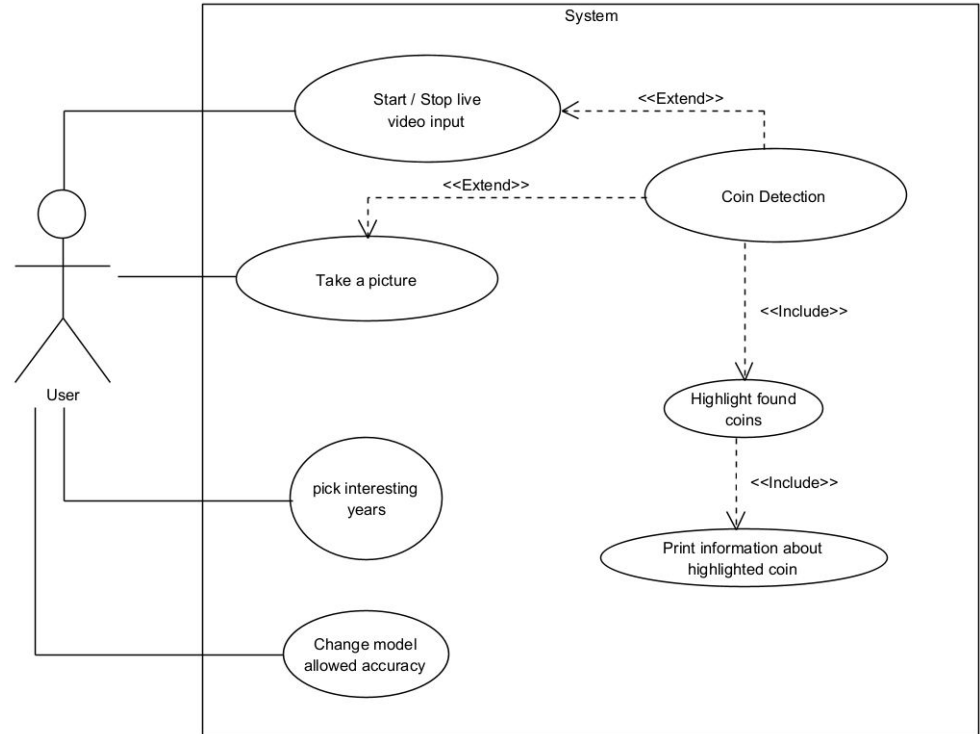
# Activity Diagram



- The following activity diagram depicts the user's interaction with the system and GUI, outlining the flow of activities and decision points

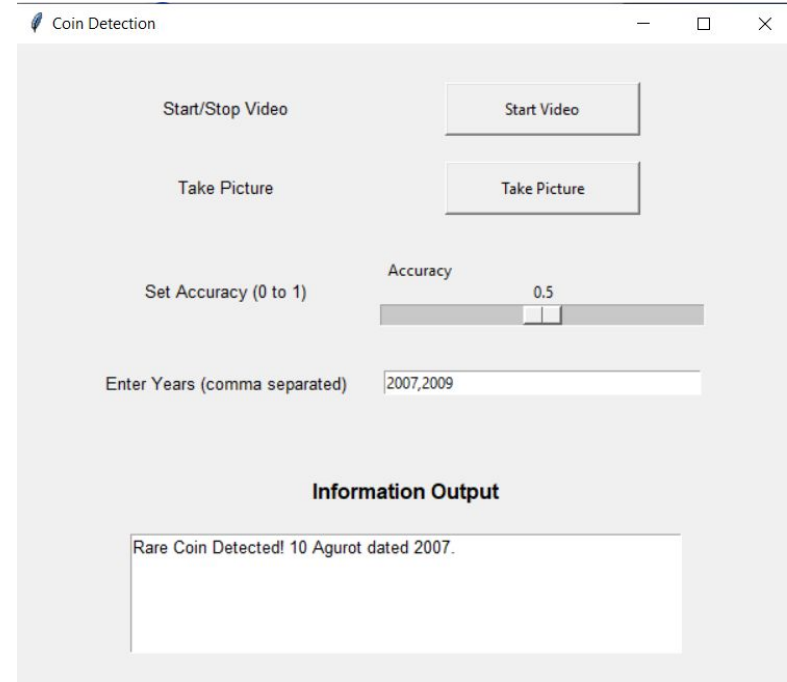
# Use case

- The following Use Case diagram shows both the user and his interaction system



# GUI

1. A button to start video mode.
2. A button to capture a photo and a button to process the given photo.
3. There will be a text box with the information after processing the coin.
4. In addition to a text box that tells the algorithm which years are needed.
5. A slider which gives the option to tell the algorithm the allowed accuracy of detection.



An example of the image after our algorithm ran on it , it highlights the coins that it found and also highlights the years of the coins that are required by the user





Thank you  
for listening!